TABLE NO

STUDENT ID NO

MULTIMEDIA UNIVERSITY

SUBJECT CODE _____

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2019/2020

TLD7011 – LOW LEVEL DESIGN OF SOFTWARE
(All sections / Groups)

23 SEPTEMBER 2019
2:30 pm – 4:30 pm
(2 Hours)

Question	Mark
A	
B	
C	
D	
Total	

Examiner 1 Signature: _____

Examiner 2 Signature: _____

Examiner 3 Signature: _____

INSTRUCTIONS TO STUDENTS

1. This question paper consists of 6 printed pages (including cover page) with 4 Sections only.

2. Attempt **ALL** questions in **SECTION A, SECTION B, SECTION C and SECTION D**. The distribution of the marks for each question is given.

3. Please write all your answers **CLEARLY** in the answer booklet provided.

Attempt ALL questions in SECTION A, B, C and D.

Section A (10 marks)

Consider the following **SingletonPatternEx.java** program.

```
// SingletonPatternEx.java
package singleton.pattern.demo;

class MakeACaptain
{
    private static MakeACaptain _captain;
    //We make the constructor private to prevent the use of "new"
    private MakeACaptain() { }

    private static class SingletonHelper{
    //Nested class is referenced after getCaptain() is called
        private static final MakeACaptain _captain = new MakeACaptain();
    } //end class SingletonHelper

    public static MakeACaptain getCaptain()
    {
        return SingletonHelper._captain;
    }
} //end class MakeACaptain

class SingletonPatternEx
{
    public static void main(String[] args)
    {
            System.out.println("***Singleton Pattern Demo***\n");
            System.out.println("Trying to make a captain for our team");
            //Put your code here
            }
}
```

Based on the above context, answer the following questions Q-A1 to Q-A3:

A1. **Rewrite the main method** of **SingletonPatternEx.java** to produce the following output by **creating two instances of singleton,** and **comparing** whether they are the same instance.

> F:\YourName>**javac singleton\pattern\demo\SingletonPatternEx.java**
>
> F:\YourName>**java singleton.pattern.demo.**SingletonPatternEx
>
> > ***Singleton Pattern Demo***
> >
> > Trying to make a captain for our team
> >
> > **Trying to make another captain for our team**
> >
> > **ct1 and ct2 are same instance**

(5 marks)

A2. **Draw** a **Class Diagram** to show TWO (2) relationships within these two classes, plus the attributes and methods.

(5 marks)

Section B (10 marks)

B1. Figure 1 shows a task dependency graph. Assuming that each task requires 1 unit of time, calculate the following metrics with workings.

(0.5 + 1 + 1 + 1 marks)

   I.   Maximum degree of concurrency
   II.   Critical path length
   III.   Total amount of work
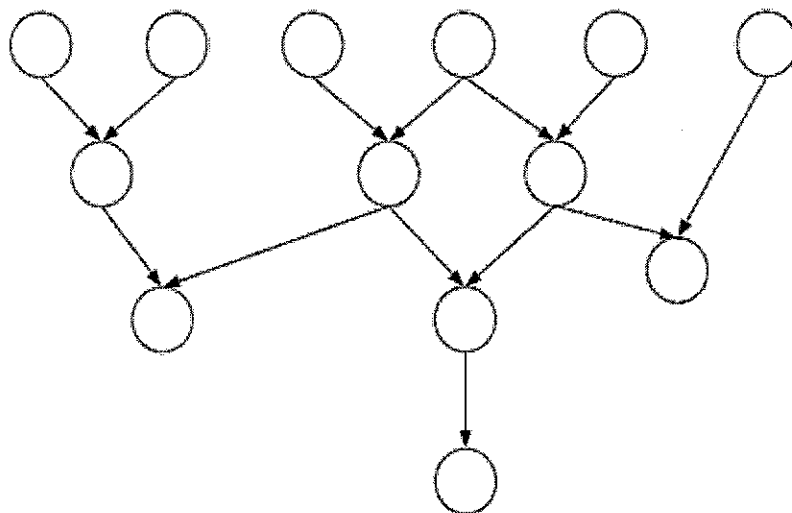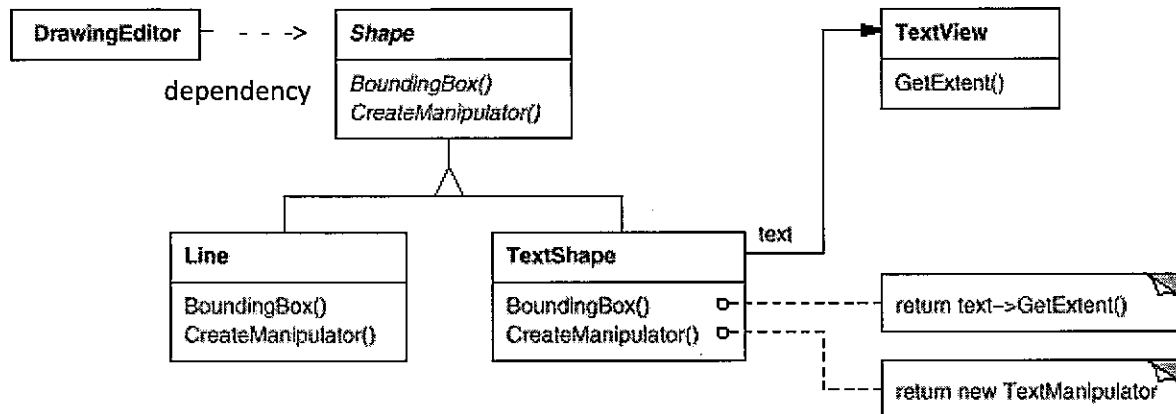   IV.   Average degree of concurrency

Fig. 1 Task Dependency Diagram

Continued...

B2. Write a table to show the mapping of the **Adapter** DP to the TextShape and TextView example given in the following class diagram. State the *Adaptee, adapteeMethod(), Target, targetMethod(), Adapter, adapteeAggregationVariable,* and *Client* in this scenario.

(4 marks)

B3. Draw a class diagram for a typical Adapter DP with *Adaptee* class, *adapteeMethod( )*, *Target* class, *targetMethod( )*, *Adapter* class, *adapteeAggregationVariable*, and *Client* class.

(2.5 marks)

Section C (10 marks)

C1. Describe the FOUR reasons why you do refactoring.

(4 marks)

C2. For each of the following codes, re-factor using the given method, pseudocode is acceptable. State the motivation to use the particular refactoring technique, and any special considerations that needs to be addressed.

(6 marks)

I. Encapsulate Field

```
1: public class Student
2: {
3:     public string FullName;
4:     public string University;
5:     public int Age;
6: }
```

Continued...

II. Extract Hierarchy

```
 1: public class Laptop
 2: {
 3:     public void installProgram()
 4:     {
 5:         // install some program
 6:     }
 7:
 8:     public void runProgram()
 9:     {
10:         // running program
11:     }
12: }
```

III. Extract Method

```
 1: public class FlightReceipt
 2: {
 3:     private IList<decimal> Discounts { get; set; }
 4:     private IList<decimal> SeatTotals { get; set; }
 5:
 6:     public decimal CalculateGrandTotal()
 7:     {
 8:         decimal subTotal = 0m;
 9:         foreach (decimal seatTotal in SeatTotals)
10:             subTotal += seatTotal;
11:
12:         if (Discounts.Count > 0)
13:         {
14:             foreach (decimal discount in Discounts)
15:                 subTotal -= discount;
16:         }
17:
18:         decimal tax = subTotal * 0.050m;
19:
20:         subTotal += tax;
21:
22:         return subTotal;
23:     }
24: }
```

Continued...

Section D (10 marks)

Generally, iterative local improvement methods are not sufficient to obtain high-quality solutions for large problems unless they are combined with other global search algorithms. This limitation has led to the creation of multilevel (or multiscale) optimization in which scale interactions of the given problem play an important role. The idea behind this is to create a hierarchy of smaller problems, which are easier to solve, and then work backward toward the solution of the original problem by using a solution inherited at the coarser level of the hierarchy to initialize the next-finer level. The hierarchy forms a basis to make global decisions for a given problem.

D1. State FIVE common problem areas that are candidates for above optimization.

(5 marks)

D2a. In inheritance, we discussed superclasses and subclasses. Which is the general class, and which is the specialized class?

D2b. What does it mean to say there is an "is a" relationship between two objects?

D2c. What does a subclass inherit from its superclass?

(1.5 marks)

D3. Explain what does each of the following command from the Java Development Kit (JDK) perform.

D3a. The *java* command.

D3b. The *javac* command.

(1 mark)

D4. Explain what does each of the following file extension means.

D4a. The *.java* file extension.

D4b. The *.class* file extension.

(1 mark)

D5. Explain THREE advantages of using the agile methodology as compared to using a traditional waterfall development methodology.

(1.5 marks)

END of Paper